

JAVA Gyakorlat

Beadandó

Neumann János Egyetem

Schiller Viktor – GWOQXX

Ferka Mihály Dávid – V7SYS6

Tartalom

1. Feladat leírás.....	2
1.1. A projekt követelményei.....	2
1.2. A Java Spring Boot technológia	3
1.3. A kezdeti lépések – adatbázis séma kivitelezése	3
2. Elrendezés kialakítása IntelliJ-ben.....	5
2.1 A felhasználói szerepek	5
3. A felhasználói szerepek.....	6
3.1.2 Adatbázis.....	9
3.1.3 Az üzenetek funkció	10
4. API funkciók.....	11
5. A Build-elés folyamata:.....	12
6. GitHub projekt:.....	13
7. Oracle felhő használata:	14

1. Feladat leírás

1.1. A projekt követelményei

Készítsen egy weboldalt egy fiktív cég, szervezet, alapítvány... részére, amely a következő követelményeknek tesz eleget:

1. Az első oldalon mutassa be a céget egy látványos weboldalon
2. Legyen Regisztráció, Bejelentkezési lehetőség
 - A „Belépés” menüpont akkor látható, ha nincs bejelentkezve a felhasználó.
 - A „Kilépés” menüpont akkor látható, ha be van jelentkezve a felhasználó.
 - A rendszer fejlécen jelenítse meg a bejelentkezett felhasználót, ha be van lépve
3. Legalább 3 felhasználói szerepet különböztessen meg:
Admin, User, Látogató
A menüpontok megjelenése és az oldalak elérhetősége változik attól függően, hogy melyik felhasználó használja az oldalt. (pl. admin oldal)
4. legyen egy oldal, ahol a választott adatbázisból jelenít meg adatokat
Ehhez 3 tábla adatait használja fel az adatbázisból.
5. Az egyik oldalon legyen egy kapcsolat űrlap, amelynek segítségével üzenetet lehet küldeni az oldal tulajdonosa számára. Ellenőrizze megfelelően az űrlap helyes kitöltését szerver oldali validációval is. Az elküldött Űrlap adatokat mentse le az adatbázisba.
6. Tegye lehetővé megtekinteni egy hatodik oldalon táblázatban az előző pontban elküldött üzeneteket az adatbázisból fordított időrend szerint (a legfrissebb legyen elől). Írja minden üzenethez a küldés idejét és az üzenetküldő nevét.
Ha nem bejelentkezett felhasználó írta, akkor: "Vendég".
7. Valósítson meg az alkalmazásban egy RESTful API-t.
Tesztelje az API funkcióit cURL-el és Postman-al is. Mindkét tesztről tegyen képernyőképeket a dokumentációba
8. Használják a GitHub (github.com) verziókövető rendszert.
(Kötelező elem! A forrás ez alapján lesz javítva)
A GitHub-on a projektmunka módszert alkalmazzák: látszódjék, hogy a csoport tagjai melyik részt készítették el és kb. fele-fele arányban járuljanak hozzá a projekthez. Ne csak a kész alkalmazást töltsék fel egy lépésben, hanem a részállapotokat is még legalább 5 lépésben személyenként.
A GIT-en saját nevet válasszanak, ami alapján be lehet azonosítani, hogy ki mit töltött fel.
Alkalmazását töltsse fel és valósítsa meg Interneten az **Oracle** felhőben.

Ezt a beadandót Schiller Viktor (GWOQXX) és Ferka Mihály Dávid (V7SYS6) készítették. A munkákat a következő hivatkozásokon lehet megtekinteni: – Github: [schillerviktor/Java_gyak_beadando \(github.com\)](https://github.com/schillerviktor/Java_gyak_beadando)– WEB: [Főoldal GWOQXX V7SYS6](#)

A bejelentkezéshez használja a következőket: – Rendszergazdaként: felhasználónév: **admin** jelszó: **admin** – Regisztrált userként a felhasználónév: **user** a jelszó: **user** – Felhasználóként nincs felhasználónév.

1.2. A Java Spring Boot technológia

A Java Spring Boot egy olyan, nyílt forráskódú eszköz, amely megkönnyíti a mikroszolgáltatások és webalkalmazások Java-alapú keretrendszerek használatával való létrehozását. A Boot bármely definíciója szerint a beszélgetésnek a Java nyelvvel kell kezdődnie, amely az alkalmazásfejlesztés egyik legnépszerűbb és legszélesebb körben használt fejlesztési nyelve és számítástechnikai platformja. A fejlesztők a világ minden táján a Java tanulásával kezdik meg a kódolás elsajátítását. A rugalmas és felhasználóbarát Java számos alkalmazás esetében a fejlesztők kedvence, a közösségi médiától, a webtől és a játékal alkalmazásoktól kezdve a hálózatkezelésig és a nagyvállalati alkalmazásokig.



1. ábra. Spring Boot design ábra

1.3. A kezdeti lépések – adatbázis séma kivitelezése

Az adatbázis séma tervezési folyamata

A tervezés az alábbi lépésekből állt:

- Az adatbázis céljának meghatározása
- A szükséges információk megkeresése és rendszerezése.

Az információk táblákra osztása

- Az információt felosztottam nagyobb entitásokra vagy témákra. Ezután majdnem minden témából tábla lesz.

Információs elemek oszlopokká alakítása

- El kellett döntenem, milyen információkat szeretnék tárolni az egyes táblákban. Minden elem mezővé alakul, és a táblában oszlopként fog megjelenni.

Elsődleges kulcsok megadása

- Meghatároztam az egyes táblák elsődleges kulcsát. Az elsődleges kulcsok az egyes sorokat egyedileg azonosító oszlopok.
- Táblakapcsolatok beállítása

- A táblákat áttekintve el kellett döntenem, hogyan kapcsolódnak az egyes táblák adatai a többi tábla adataihoz. Ha egyértelművé szeretném tenni a táblák kapcsolatát, létrehozhatok új mezőket a táblákban, vagy létrehozhatok új táblákat.

A terv finomítása

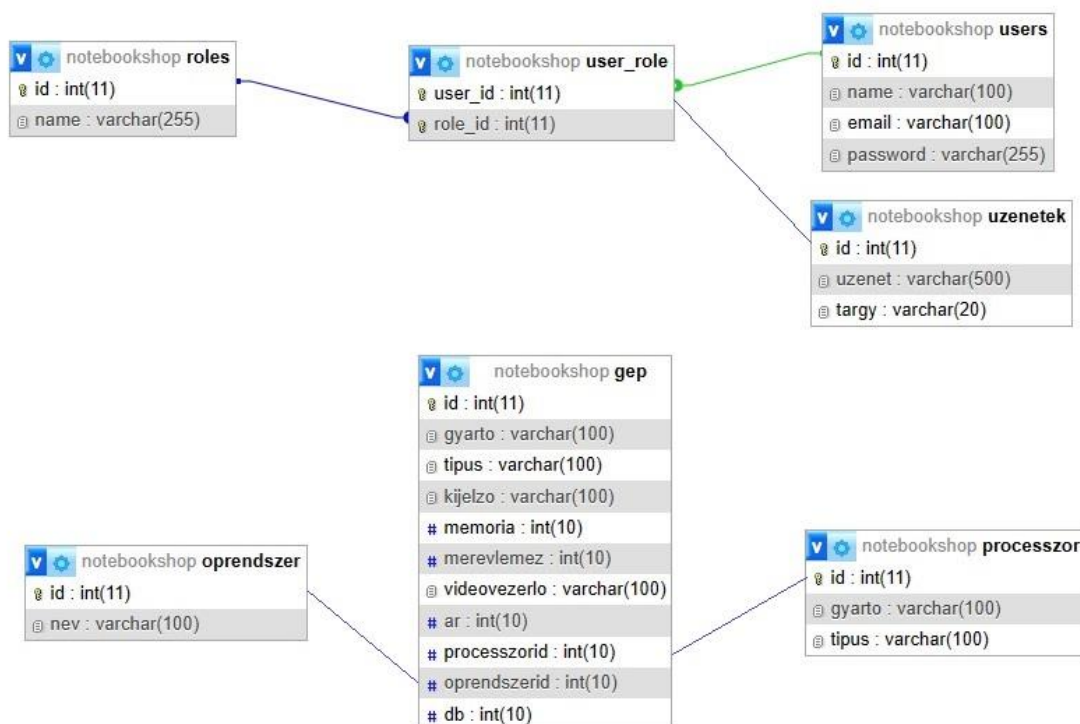
- Meggyőződtem róla, hogy hibáktól mentes a terv. Létrehoztam a táblákat, és hozzáadtam néhány rekordnyi mintaadatot. Ellenőriztem, hogy a táblák a kívánt eredményeket adják-e. Szükség esetén módosítani kellett a tervet.

A normalizációs szabályok alkalmazása

- Az adatnormalizációs szabályokat alkalmazva ellenőriztem, hogy a táblák szerkezete megfelelő-e. (1NF, 2NF, 3NF, stb.)

Az információk táblákba való felosztásának egyik stratégiája az, hogy szemügyre vesszük az egyes dolgokat és meghatározzuk, hogy ezek mire vonatkoznak. Keressük meg azokat az információkat, amelyekhez az adatbázisból hozzá kívánunk férni, és azt osszuk fel olyan témakörökre, amelyeket nyomon kívánunk követni! Ezen témakörök mindegyikéhez szükséges lehet egy-egy külön tábla.

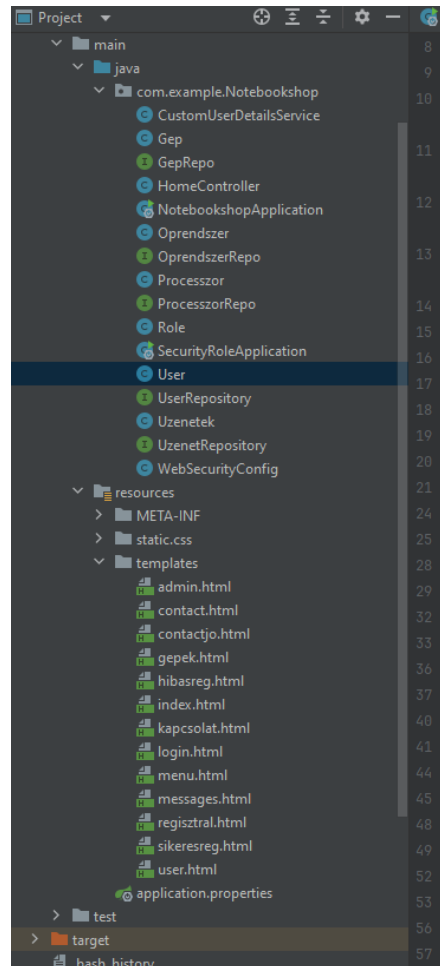
A fenti terv alapján az alábbi adatmodell került kialakításra:



2. ábra. A notebookshop adatmodell

2. Elrendezés kialakítása IntelliJ-ben

Az új Projektet IntelliJ-ből indítottam, ennek hatására létre jön a munka és az összes szükséges fájl ami egy Java alkalmazás működéséhez szükséges.



4. ábra. IntelliJ-ben létrejövő fájl struktúra

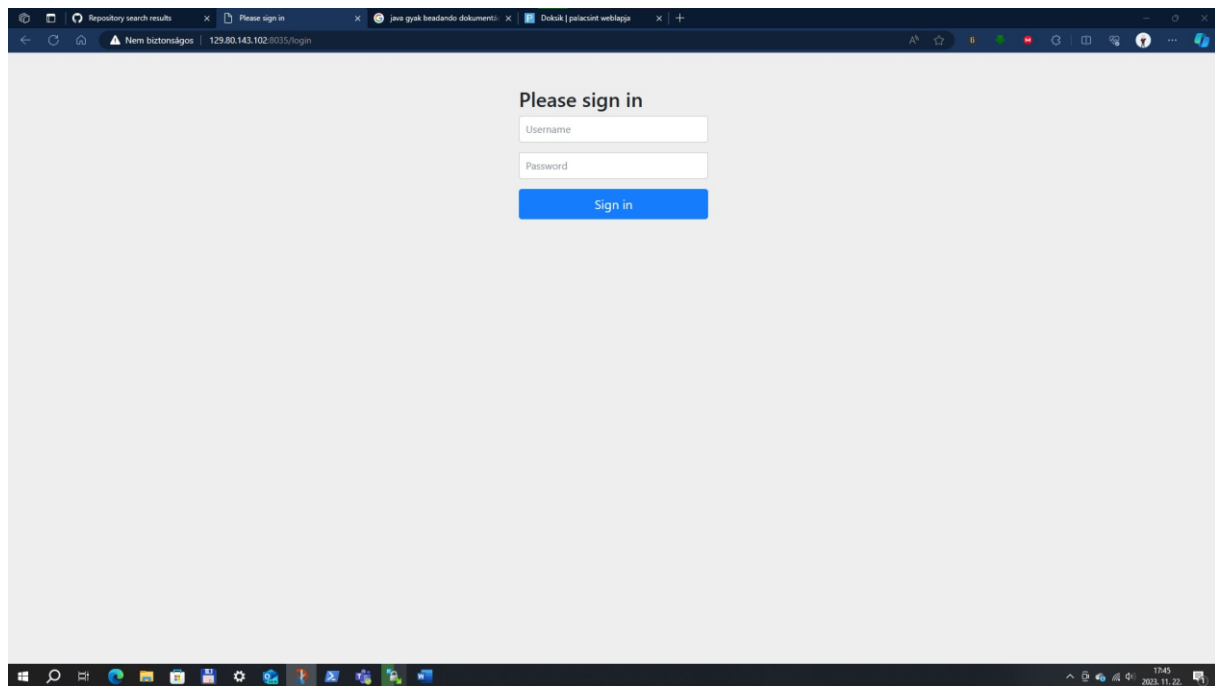
2.1 A felhasználói szerepek

Legyen Regisztráció, Bejelentkezési lehetőség.

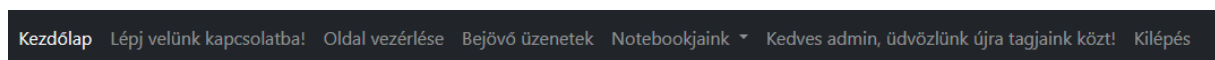
A „Belépés” menüpont akkor látható, ha nincs bejelentkezve a felhasználó.

- A „Kilépés” menüpont akkor látható, ha be van jelentkezve a felhasználó.

- A rendszer fejlécen jelenítse meg a bejelentkezett felhasználót, ha be van lépve.



5. ábra. Login képernyő a látogatóknak



6. ábra. 'Kilépés' opció a már bejelentkezett felhasználóknak

3. A felhasználói szerepek

Legalább 3 felhasználói szerepet különböztessen meg:

Admin, User, Látogató

A menüpontok megjelenése és az oldalak elérhetősége változik attól függően, hogy melyik felhasználó használja az oldalt. (pl. admin oldal)

```
@Entity
@Table(name="users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    private String name;
    private String email;
    private String password;
    @ManyToMany(cascade=CascadeType.MERGE)
    @JoinTable(
        name="user_role",
        joinColumns={@JoinColumn(name="USER_ID",
referencedColumnName="ID")},
        inverseJoinColumns={@JoinColumn(name="ROLE_ID",
referencedColumnName="ID")})
}
```



```

private List<Role> roles;

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public List<Role> getRoles() {
    return roles;
}

public void setRoles(List<Role> roles) {
    this.roles = roles;
}

```

9. ábra. A User.java osztály

```

package com.example.Notebookshop;
import javax.persistence.*;
import java.util.List;

@Entity
@Table(name = "roles")
public class Role {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(nullable = false, unique = true)
    private String name;
    @ManyToMany(mappedBy = "roles")
    private List< User > users;
}

```



```

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getName() {
    return name;
}

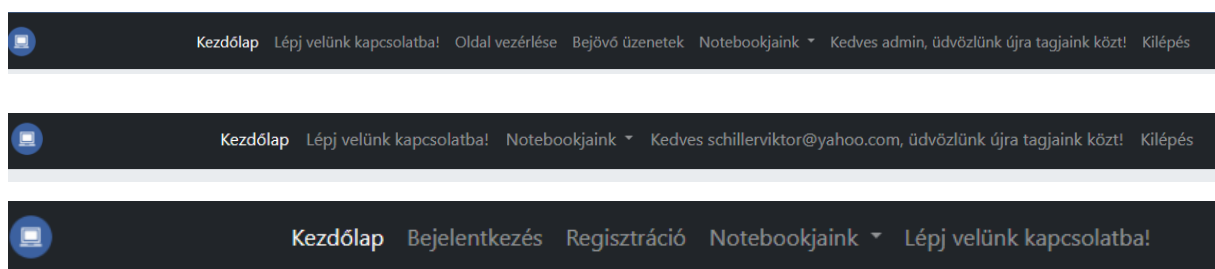
public void setName(String name) {
    this.name = name;
}

public List<User> getUsers() {
    return users;
}

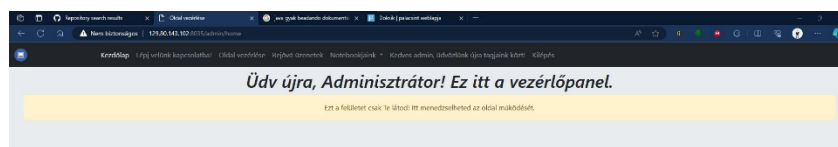
public void setUsers(List<User> users) {
    this.users = users;
}
}

```

10. ábra. A Role.java osztály



11. ábra. Az 'admin', a regisztrált felhasználó és a látogató bejelentkezés fejléce



12. ábra. Az 'admin' adminisztrátori oldala

3.1.2 Adatbázis

legyen egy oldal, ahol a választott adatbázisból jelenít meg adatokat, ehhez 3 tábla adatait használja fel az adatbázisból.

A következő konfigurációk közül tudsz választani:

Gyártó és típus	Memória mérete	HDD mérete	Videóvezérlő	Ára
HP COMPAQ 615 NX556EA	1024 MB	160 GB	ATI Mobility Radeon HD3200 256MB	95120 Ft
ASUS K51AC-SX001D	2048 MB	250 GB	ATI Mobility Radeon HD3200 256MB	101200 Ft
HP COMPAQ 615 NX560EA	2048 MB	320 GB	ATI Mobility Radeon HD3200 256MB	114800 Ft
HP Pavilion DV6-1110EH NL956EA	3072 MB	250 GB	ATI Mobility Radeon HD4530 512MB	167920 Ft
ACER Aspire 5536G-64G2G3MN	2048 MB	250 GB	ATI Mobility Radeon HD4570 512MB	111920 Ft
Acer Aspire 5536G-64G3G3MNN	3072 MB	320 GB	ATI Mobility Radeon HD4570 512MB	117520 Ft
MSI X410-019HU	2048 MB	320 GB	ATI Radeon Xpress 1280	111920 Ft
ASUS E61T-VX000X	4096 MB	500 GB	ATI Mobility Radeon HD4870 512MB	115920 Ft
MSI VR320X-004IU	2048 MB	320 GB	NVIDIA GeForce Go 9100M-GS	90800 Ft
ASUS N60DP-IX012V	4096 MB	500 GB	ATI Mobility Radeon HD4670 512MB	183920 Ft
ASUS K50AB-SX04SD	4096 MB	500 GB	ATI Mobility Radeon HD4570 512MB	134320 Ft
FUJITSU Amilo Sa3650	2048 MB	250 GB	ATI Mobility Radeon HD3200 256MB	223920 Ft
MSI WIND U200-064HU	2048 MB	320 GB	Intel Graphics X4500M / 256MB	112400 Ft
ACER eMachine E525-901G16MI	1024 MB	160 GB	Intel Graphics 4500MHd	82800 Ft
DELL Inspiron 1545-106208 RED	1024 MB	160 GB	Intel Graphics 4500MHd	103120 Ft
TOSHIBA Satellite L500-1EN	2048 MB	320 GB	Intel Graphics X4500M / 256MB	118800 Ft
MSI CR500X-012HU	2048 MB	320 GB	NVIDIA GeForce Go 8200M 128MB	94800 Ft
MSI CR500X-008HU	4096 MB	320 GB	NVIDIA GeForce Go 8200M 128MB	95920 Ft
LENOVO SL500 2746PSG	1024 MB	160 GB	NVIDIA GeForce Go 9300M 256MB	139920 Ft
TOSHIBA Satellite L300-24P	2048 MB	160 GB	Intel Graphics 4500MHd	96000 Ft
MSI VR320X-004HU	4096 MB	320 GB	Intel Graphics 4500MHd	89600 Ft
HP ProBook 4510s NX435EA	2048 MB	250 GB	Intel Graphics 4500MHd	111920 Ft
FUJITSU Espimo V635s	1024 MB	160 GB	Intel Graphics 4500MHd	95920 Ft
LENOVO IdeaPad G550L 59 026377	1024 MB	160 GB	Intel Graphics 4500MHd	94320 Ft
HP Presario CQ61-200SH NZ980EA	3072 MB	320 GB	Intel Graphics 4500MHd	127120 Ft
ACER eMachine E525-302G25MI	2048 MB	250 GB	Intel Graphics 4500MHd	89200 Ft
HP ProBook 4510s NX068EA	2048 MB	250 GB	Intel Graphics 4500MHd	113520 Ft
HP ProBook 4310s NX808EA	1024 MB	160 GB	Intel Graphics 4500MHd	119920 Ft
ASUS K50U-SX036L	2048 MB	250 GB	Intel Graphics X4500M	94320 Ft
ASUS K50U-SX153L	4096 MB	320 GB	Intel Graphics X4500M / 256MB	100720 Ft
MSI CR700X-023HU	3072 MB	320 GB	NVIDIA GeForce Go 8200M 128MB	108400 Ft
DELL Vostro V860-111696	1024 MB	250 GB	Intel Graphics x3100	79920 Ft
HP Mini 1199 NX528EA	1024 MB	80 GB	Intel Graphics 950	116000 Ft

13. ábra. Az adatbázis adataira történő hivatkozás és betöltés funkciója

Az adatbázis kialakítását SSH-n keresztül egy iskolai Oracle felhőben futtatott Linux alapú rendszerben hoztuk létre. A megadott adatbázisok elérése után ki kellett adni az SQL parancsokat:

```
mysql> type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use adasthai089;
Database changed
mysql> [adasthai089]: CREATE DATABASE IF NOT EXISTS `notebookshop` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
mysql> {
    { id(int(11) NOT NULL auto_increment,
      name varchar(100) NOT NULL,
      email varchar(100) NOT NULL,
      password varchar(255) NOT NULL,
      PRIMARY KEY (`id`))
    } ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `users` (`id`,`name`,`email`,`password`) VALUES
(1,'admin','admin','$1d6M5IrxwBPtKerxvN/DXMS/OFFML2mPQPyCuF3j2acWqHTBz4l81'),
(2,'user','user','$1d6M5IrxwBPtKerxvN/DXMS/OFFML2mPQPyCuF3j2acWqHTBz4l81');

-----
CREATE TABLE `roles` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `roles` (`id`,`name`) VALUES
(1,'halg089@glocalhost') to database `notebookshop`
mysql> [adasthai089]: USE `notebookshop`;
mysql> SHOW GRANTS FOR user@% ; access denied for user 'halg089@glocalhost' to database `notebookshop`
mysql> [adasthai089];
mysql> [adasthai089]: CREATE TABLE `users` (
->   `id` int(11) NOT NULL auto_increment,
->   `name` varchar(100) NOT NULL,
->   `email` varchar(100) NOT NULL,
->   `password` varchar(255) NOT NULL,
->   PRIMARY KEY (`id`)
->) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
);
id, `ROLE_USER`);

-----
CREATE TABLE `user_role` (
  `user_id` int(11) NOT NULL,
  `role_id` int(11) NOT NULL,
  PRIMARY KEY (`user_id`,`role_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `user_role` (`user_id`,`role_id`) VALUES
(1,1),
(1,2);

CREATE TABLE `emp` (
  `id` int(11) NOT NULL auto_increment PRIMARY KEY,
  `gender` varchar(100) NOT NULL,
  `title` varchar(100) NOT NULL,
  `kijelo` varchar(100) NOT NULL,
  `memois` int(10) NOT NULL,
  `reviewer` int(10) NOT NULL,
  `idosecepr` varchar(100) NOT NULL,
  `ar` int(10) NOT NULL,
  `processstop` int(10) NOT NULL,
  `opendescrict` int(10) NOT NULL,
  `cd` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

14. ábra. 'Create Table' parancsok a felhőben

```

MariaDB [adatbazis027]> COMMIT;
Query OK, 0 rows affected (0.000 sec)

MariaDB [adatbazis027]> show tables;
+-----+
| Tables_in_adatbazis027 |
+-----+
| gep                     |
| oprendszerek            |
| processzor              |
| roles                   |
| user_role               |
| users                   |
| uzenetek                 |
+-----+
7 rows in set (0.000 sec)

MariaDB [adatbazis027]>

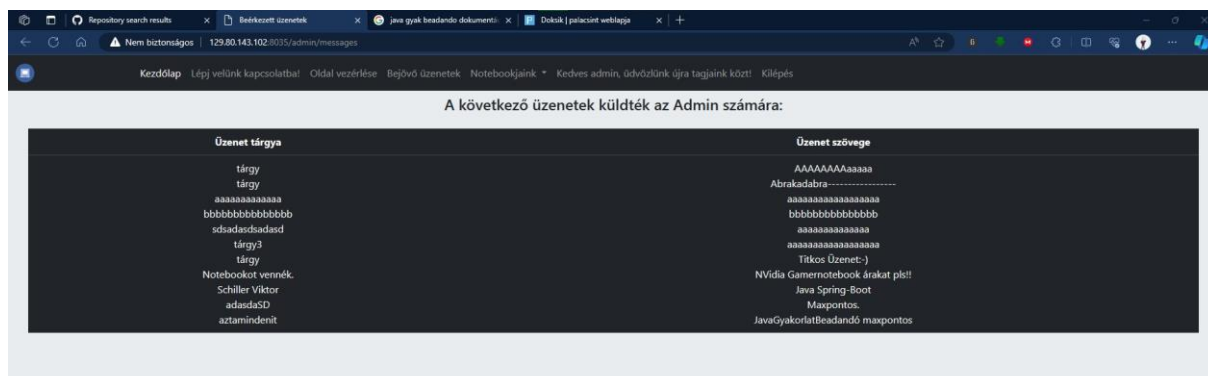
```

15. ábra. A táblák lekérdezése

3.1.3 Az üzenetek funkció

Az egyik oldalon legyen egy kapcsolat űrlap, amelynek segítségével üzenetet lehet küldeni az oldal tulajdonosa számára. Ellenőrizze megfelelően az űrlap helyes kitöltését szerver oldali validációval is. Az elküldött Űrlap adatokat mentse le az adatbázisba. Ezenkívül tegye lehetővé megtekinteni egy hatodik oldalon táblázatban az előző pontban elküldött üzeneteket az adatbázisból fordított időrend szerint (a legfrissebb legyen elől). Írja minden üzenethez a küldés idejét és az üzenetküldő nevét. Ha nem bejelentkezett felhasználó írta, akkor: "Vendég".

16. ábra. Az üzenetek beviteli mezői



17. ábra. Az üzenetek megjelenítése 'admin' módban

Az üzenetek természetesen tárolva lesznek az adatbázis „uzenetek” táblájában és az adminisztrátor tudja olvasni azokat.

4. API funkciók

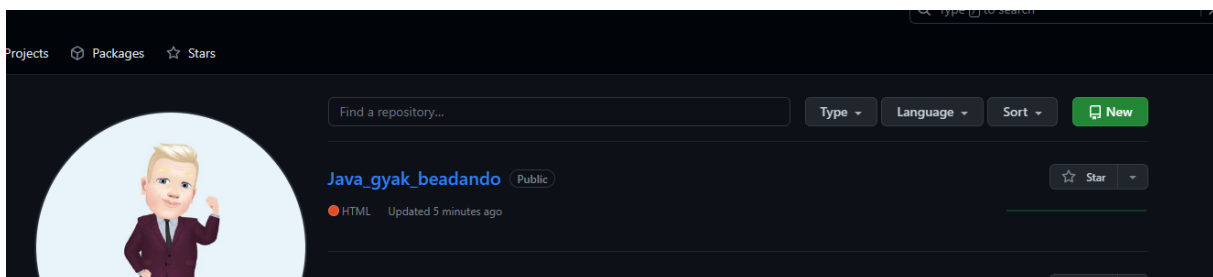
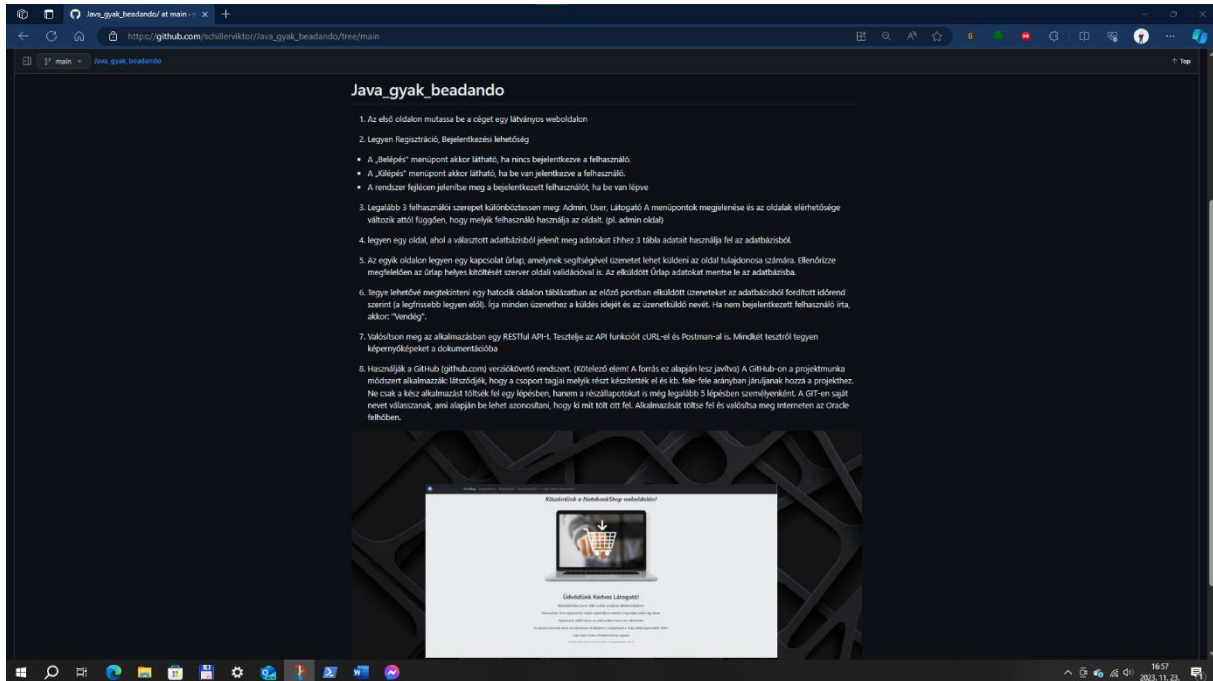
Valósítson meg az alkalmazásban egy RESTful API-t. Tesztelje az API funkcióit cURL-el és Postman-al is. Mindkét tesztről tegyen képernyőképeket a dokumentációba.



18. ábra. YouTube beágyazott API

6. GitHub projekt:

Használtuk a GitHub (github.com) verziókövető rendszert, mrgosztva egymással a tapasztalatokat és akadályokat.



20. ábra. GitHub verziókezelő használata

7. Oracle felhő használata:

Alkalmazását töltsse fel és valósítsa meg Interneten az Oracle felhőben. Az URL címében szerepeljen a csoport egyik tagjának a neve.

```
[hallgato027@instance-20230904-1042 ~]$ ls -l
total 44
-rw-rw-r--. 1 hallgato027 hallgato027 13089 Oct 27 16:09 demo.iml
-rw-rw-r--. 1 hallgato027 hallgato027 10284 Jun 28 2022 mvnw
-rw-rw-r--. 1 hallgato027 hallgato027 6734 Jun 28 2022 mvnw.cmd
-rw-rw-r--. 1 hallgato027 hallgato027 3308 Nov 5 20:53 pom.xml
drwxrwxr-x. 4 hallgato027 hallgato027 30 Nov 14 17:19 src
drwxrwxr-x. 9 hallgato027 hallgato027 4096 Nov 14 17:46 target
[hallgato027@instance-20230904-1042 ~]$
```

21. ábra. Oracle Linux OP használata, a feltöltött fájlrendszer

```
[hallgato035@instance-20230904-1042 ~]$ mysql -u hallgdb035 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 230
Server version: 10.3.35-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use adatbazis035
Database changed
MariaDB [adatbazis035]>
```

22. ábra. Oracle Linux OP használata, adatbázis elérése